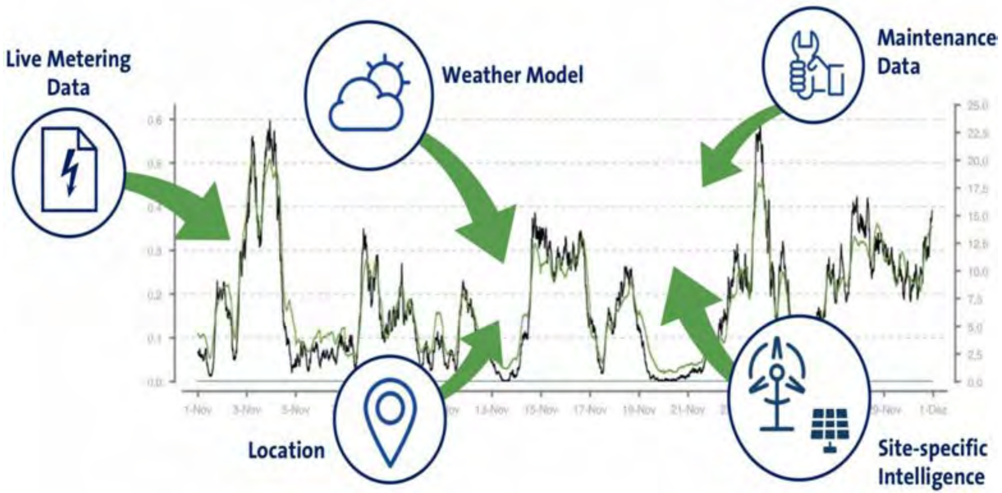
Enhancing Energy Management: Advanced Techniques for Forecasting and Optimization

Mahim Rahaman, REU Scholar (Lehman college)
Under the guidance of Dr. Zhen Ni

NSF REU IN SENSING AND SMART SYSTEMS – FAU 2024
Infrastructure Systems: Machine Learning Techniques for Energy Forecasting and Optimization

Background



Problem:

High energy consumption causes higher power bills, environmental harm, depletion of resources and infrastructure strain.

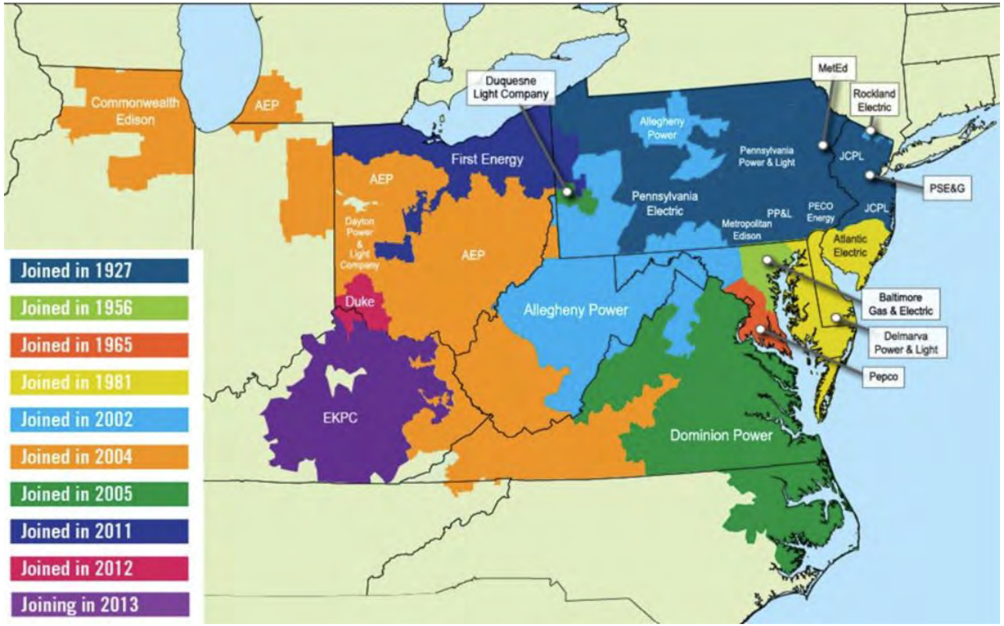
Solution:

We use machine learning models to predict energy consumption trends, optimize usage, reduce costs, and support renewable energy integration.

Existing solution:

Lack accuracy, adapting to changes, addition of new data and flexibility.

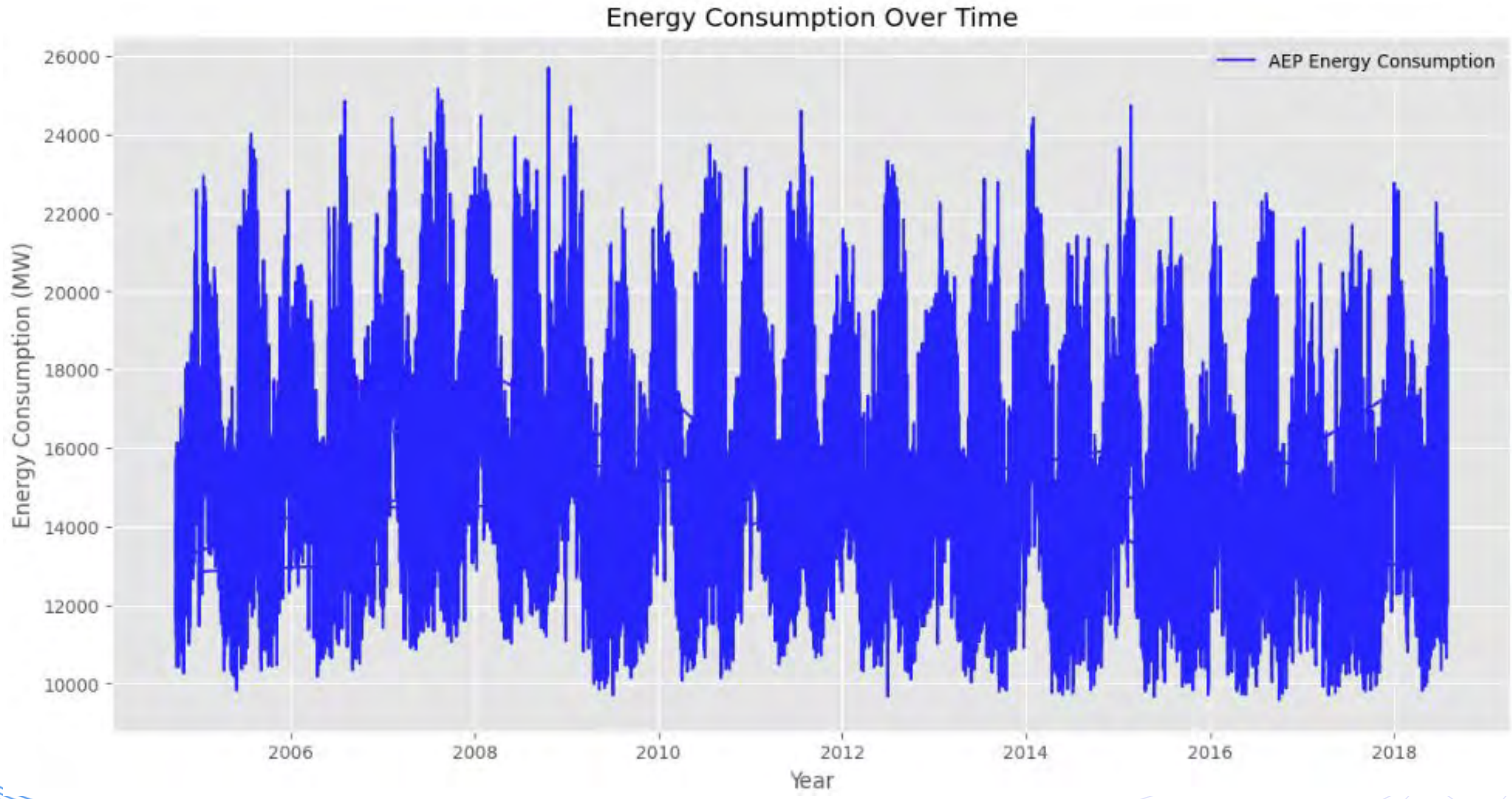
Data set



American Electric power (AEP) produces energy throughout the United States which include Arkansas, Indiana, Kentucky, Louisiana, Michigan, Ohio, Oklahoma, Tennessee, Texas, Virginia and West Virginia.

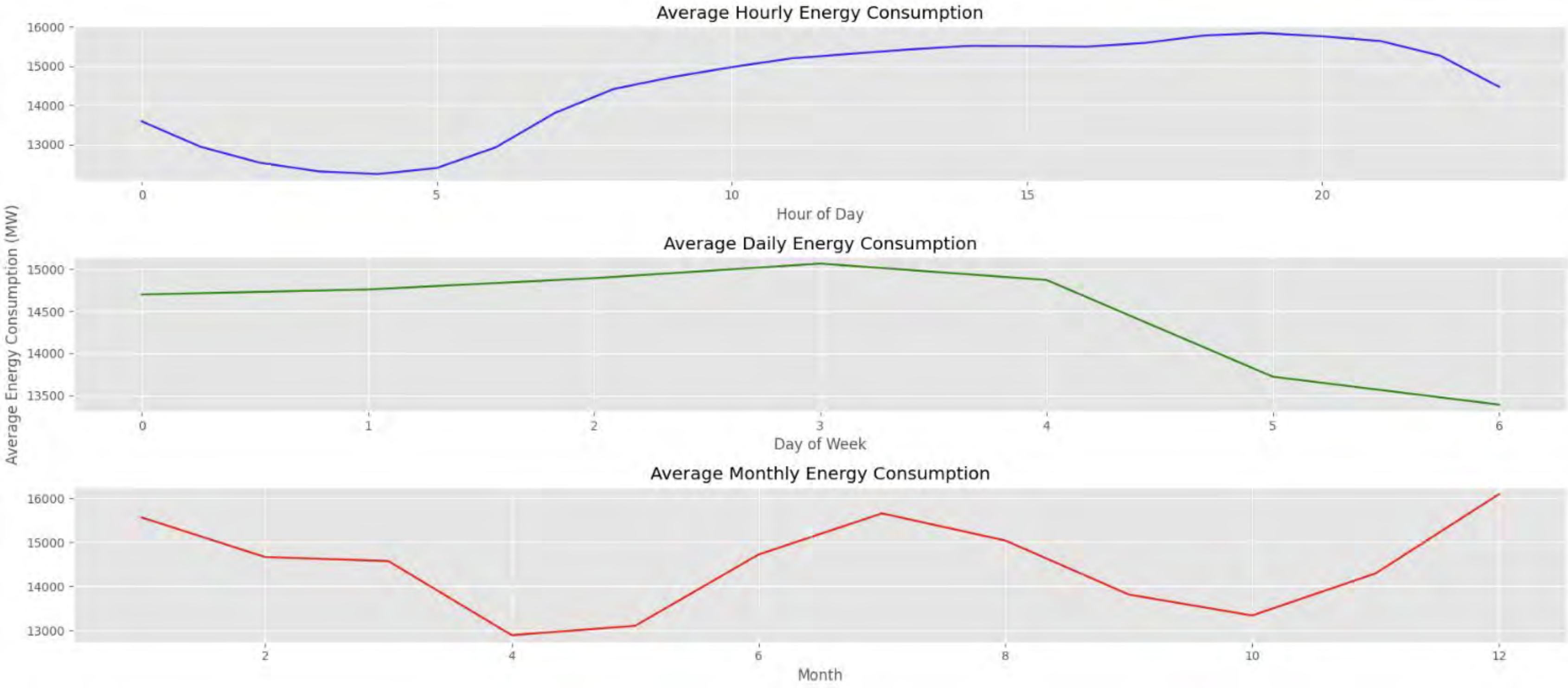
https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption

Visualization of the Data



Analysis of data trends

Start Date : 2017-01-01
End Date: 2017-12-31



Data Split

Training Set Preview:

Datetime	AEP_MW	hour	day_of_week	month
2017-01-01 00:00:00	13240.0	0	6	1
2017-01-01 01:00:00	12876.0	1	6	1
2017-01-01 02:00:00	12591.0	2	6	1
2017-01-01 03:00:00	12487.0	3	6	1
2017-01-01 04:00:00	12369.0	4	6	1

Testing Set Preview:

Datetime	AEP_MW	hour	day_of_week	month
2017-10-01 00:00:00	10948.0	0	6	10
2017-10-01 01:00:00	10460.0	1	6	10
2017-10-01 02:00:00	10060.0	2	6	10
2017-10-01 03:00:00	9960.0	3	6	10
2017-10-01 04:00:00	9835.0	4	6	10

Splitting DATA 75/25

Training : Jan 2017 - Sept 2017

Testing: Oct 2017 - Dec 2017

Model Planning for LSTM and CNN

Shape: [time steps, features]

Input: [24, 1]

→ **Model** →

Output: [Sample, 1]
MW per sequence

Training and results Long Short-Term Memory (LSTM) model

```
Model: "sequential"
_________________________________________________________________
 Layer (type)              Output Shape              Param #
=================================================================
 lstm (LSTM)               (None, 50)                10400

 dense (Dense)             (None, 1)                 51

=================================================================
Total params: 10451 (40.82 KB)
Trainable params: 10451 (40.82 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Root Mean Squared Error: 343.2 MW

Train MSE: 133,963.17 MW^2

Test MSE: 117,843.34 MW^2

Mean Absolute Percentage Error (MAPE): 1.574%

Training and results convolutional neural network (CNN) model

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 22, 64)            256

 max_pooling1d (MaxPooling1  (None, 11, 64)            0
 D)

 flatten (Flatten)           (None, 704)               0

 dense (Dense)               (None, 50)                35250

 dense_1 (Dense)             (None, 1)                 51

=================================================================
Total params: 35557 (138.89 KB)
Trainable params: 35557 (138.89 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

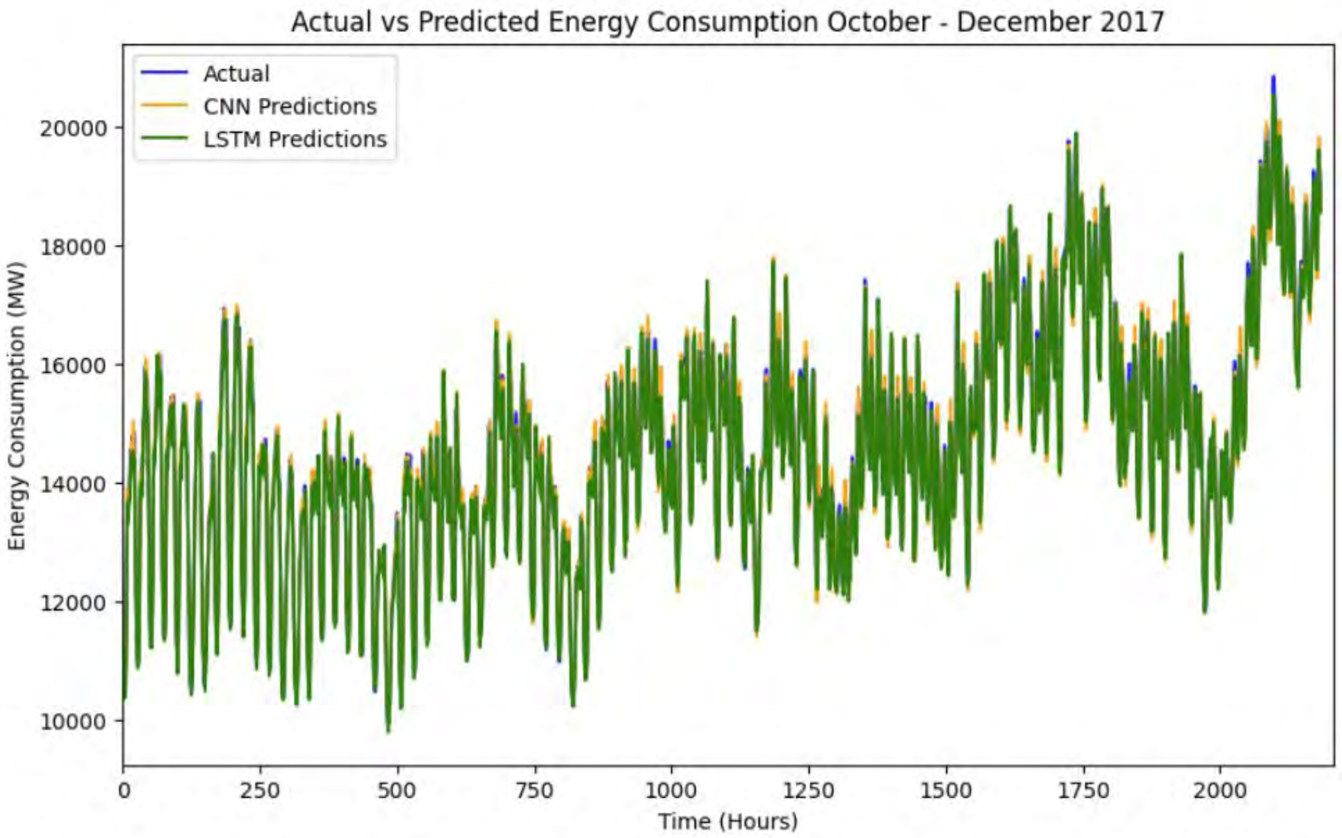
Root Mean Squared Error: 320.37 MW

Train MSE: 133,963.17 MW^2

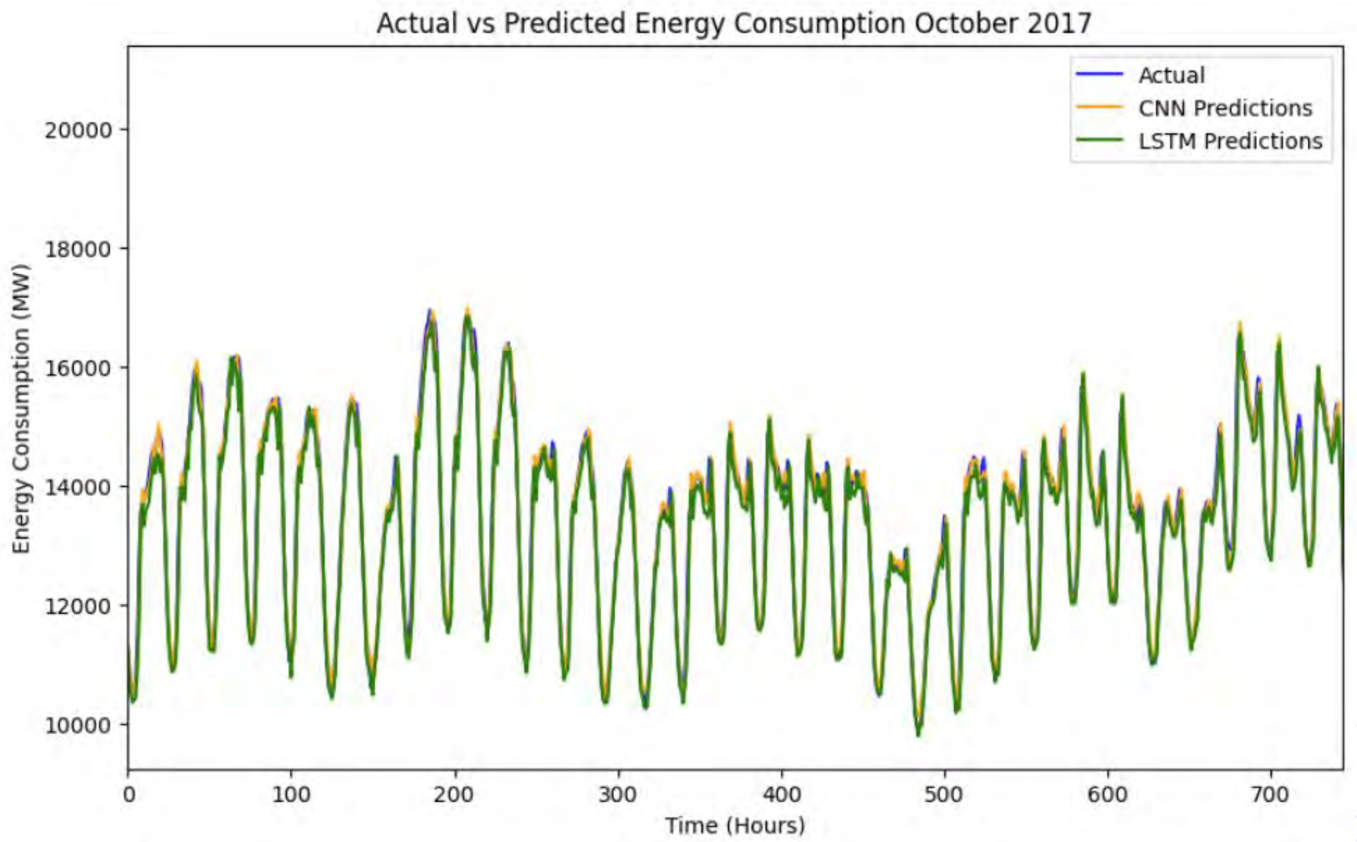
Test MSE: 102,640.75 MW^2

Mean Absolute Percentage Error (MAPE): 1.741%

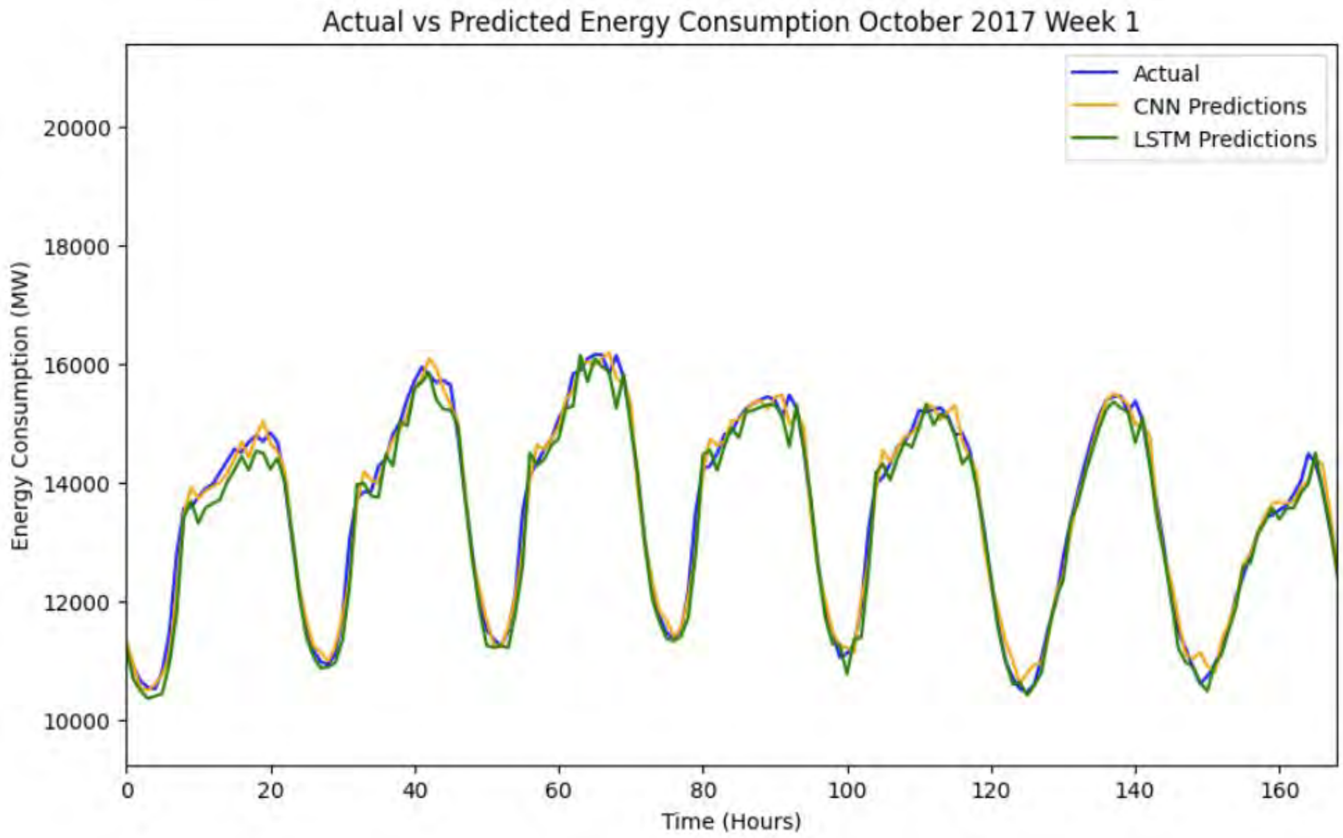
Result analysis



Result analysis



Result analysis



Transformer model

A Transformer model is a type of machine learning model that is especially good at understanding and generating sequences of data, like sentences. It uses a mechanism called "attention" to focus on different parts of the input data as needed, rather than processing it in order from start to finish.

- Encoder: Takes the input data (like a sentence in one language) and turns it into a set of features or representations.
- Decoder: Takes these features and generates the output data (like a translated sentence in another language).

Model planning

Shape: [time steps, features]

Input: [24, 1]

Model

Output: [Sample, 1]

N-steps which in this case is 24

1 -Feature ‘AEP_MW’

Sample - Target value

1 - per sequence

All 3 model planning

Transformer

Root Mean Squared Error: 57.61 MW

MSE: 3319.29 MW^2

Mean Absolute Percentage Error (MAPE): 0.0032%

LSTM

Root Mean Squared Error: 343.2 MW

MSE: 117,843.34 MW^2

Mean Absolute Percentage Error (MAPE): 1.574%

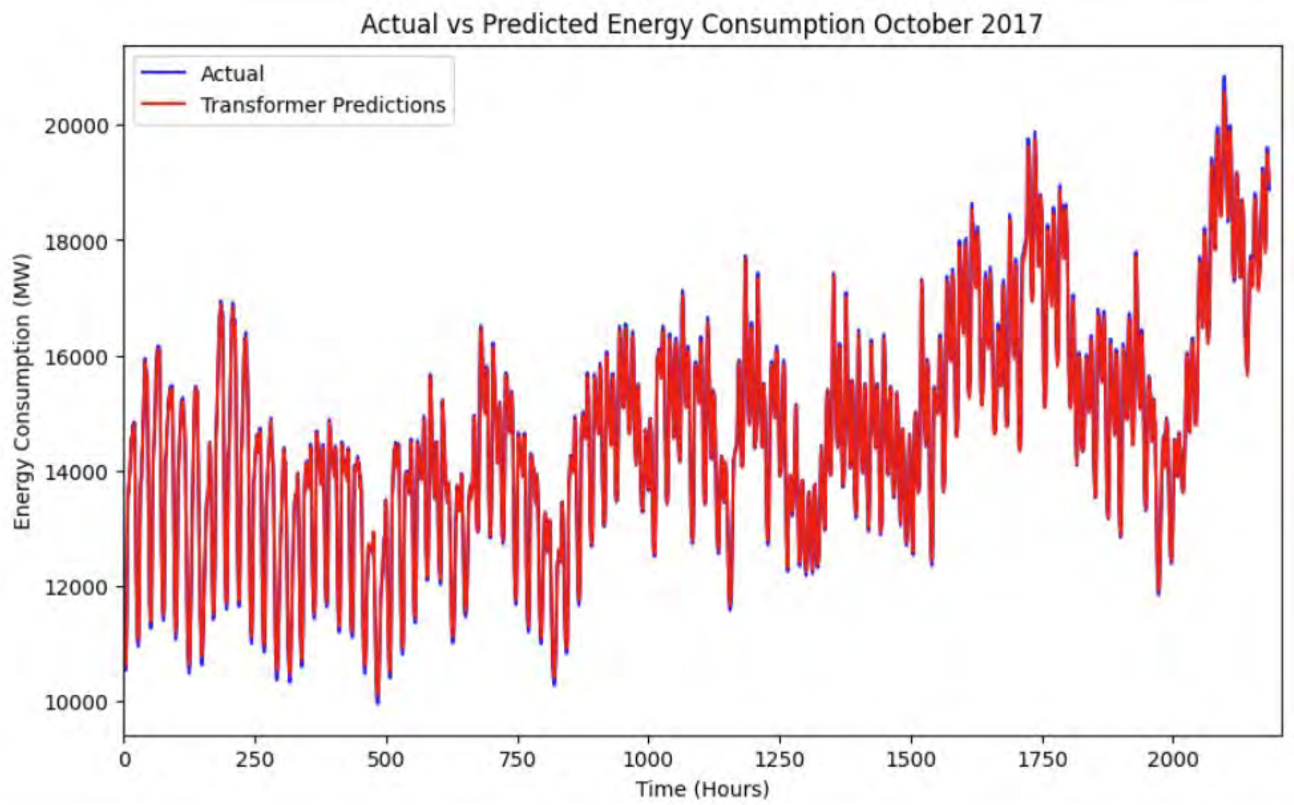
CNN

Root Mean Squared Error: 320.37 MW

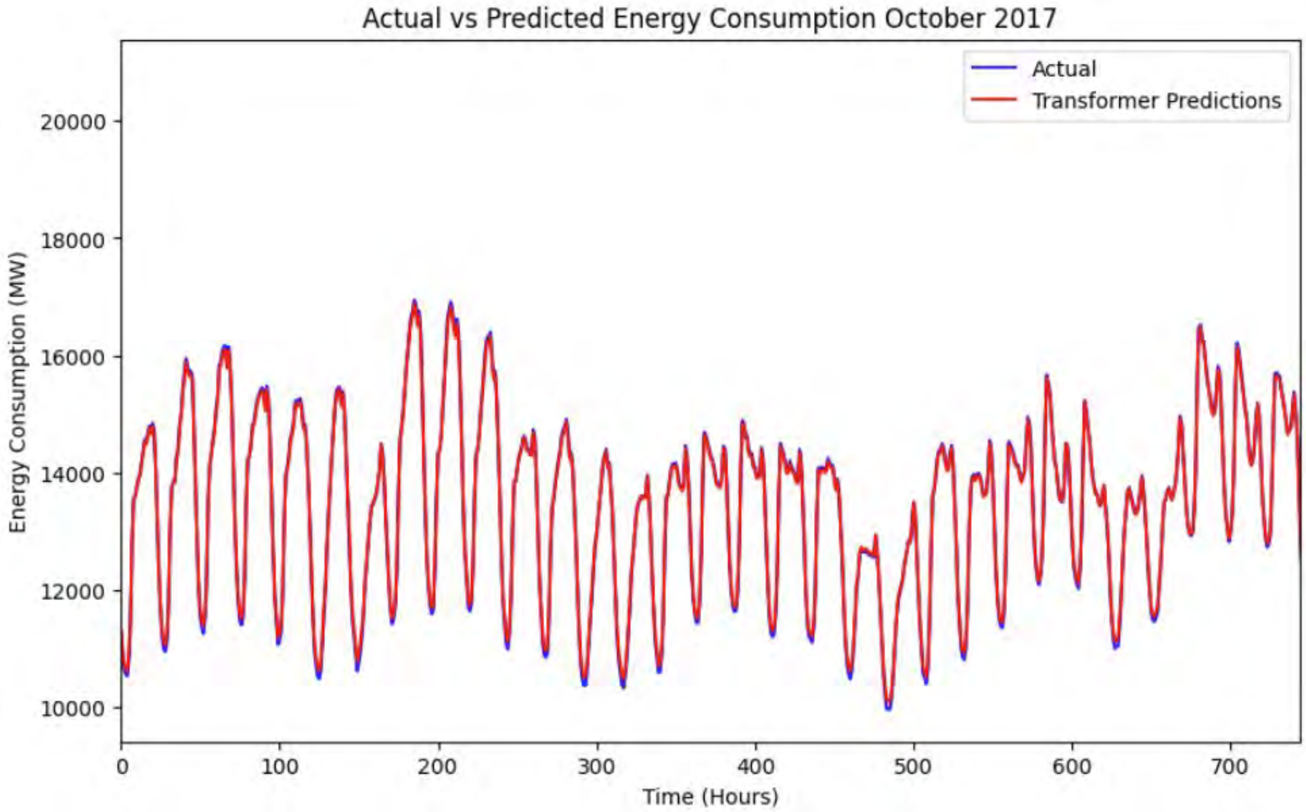
MSE: 102,640.75 MW^2

Mean Absolute Percentage Error (MAPE): 1.741%

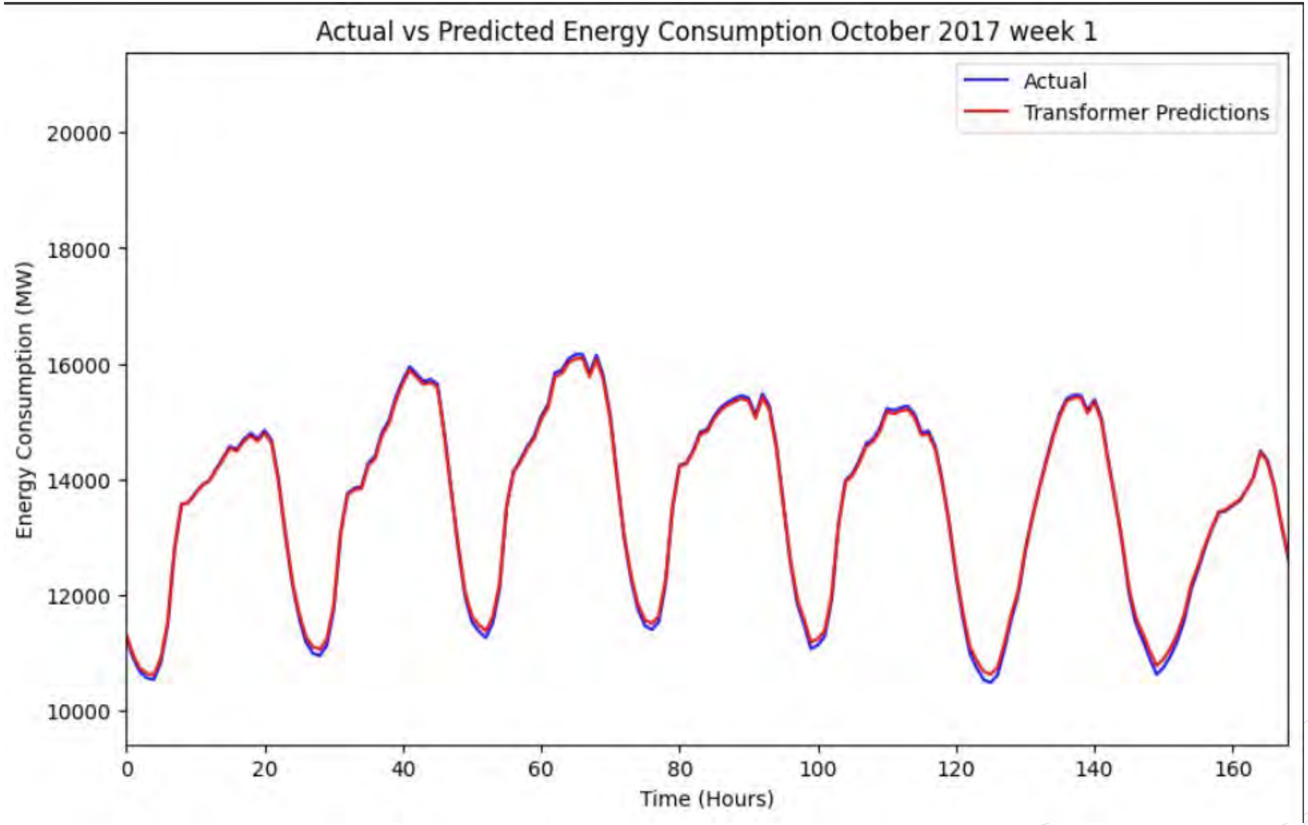
Visualization of the Transformer model



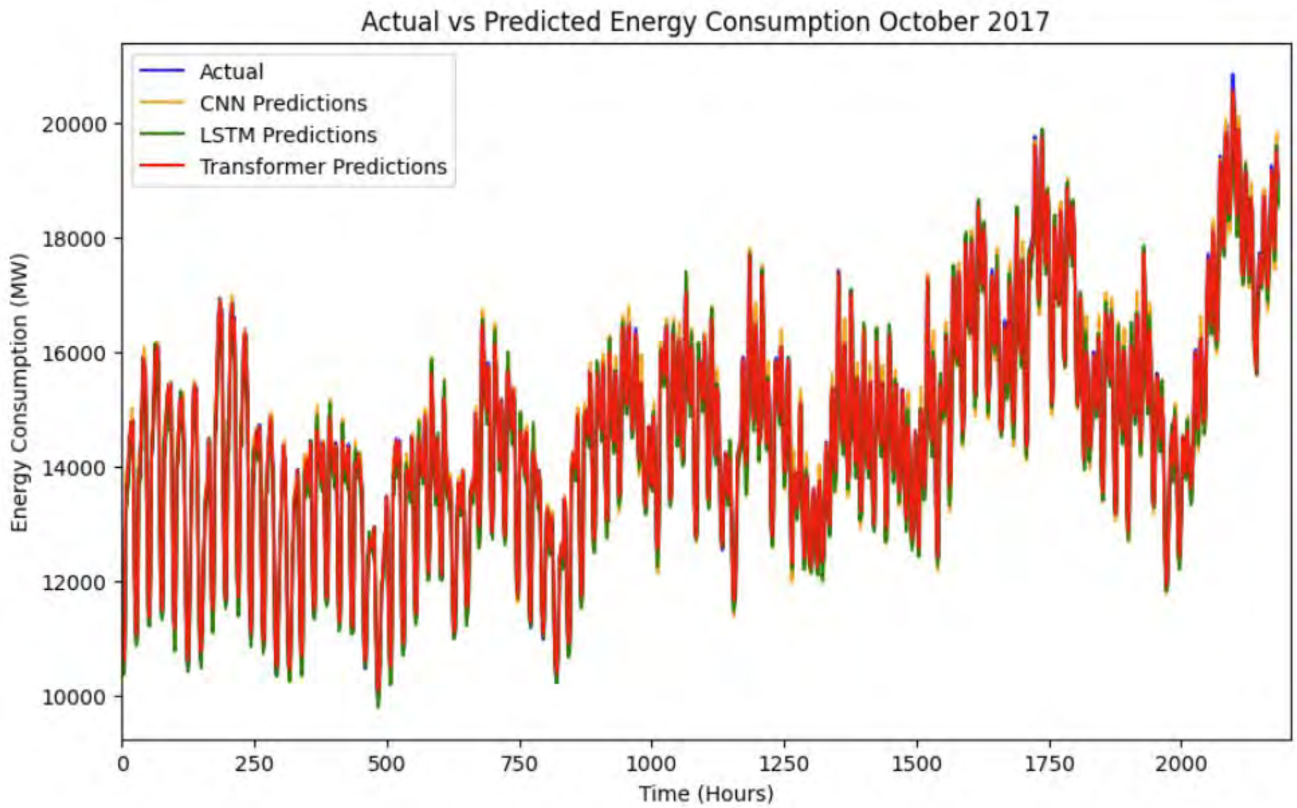
Visualization of the Transformer model



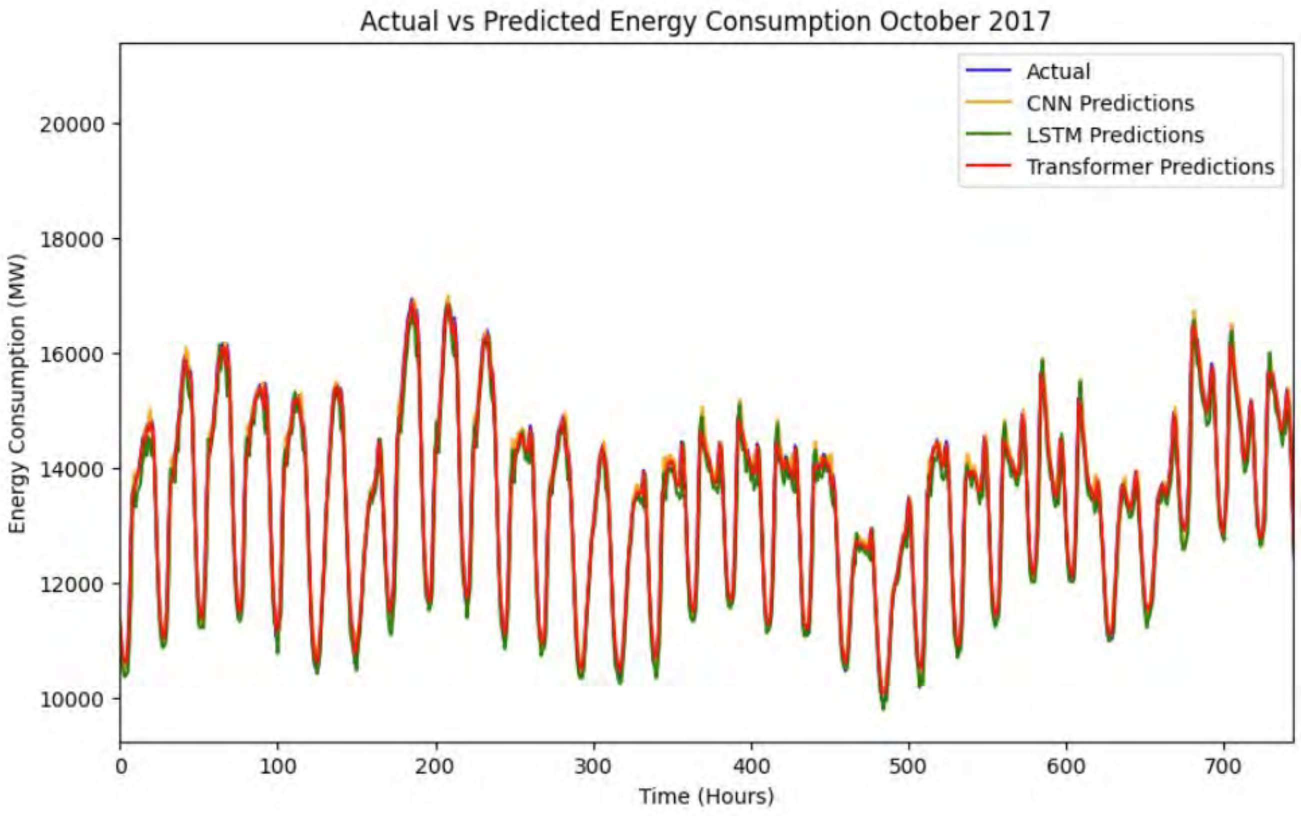
Visualization of the Transformer model



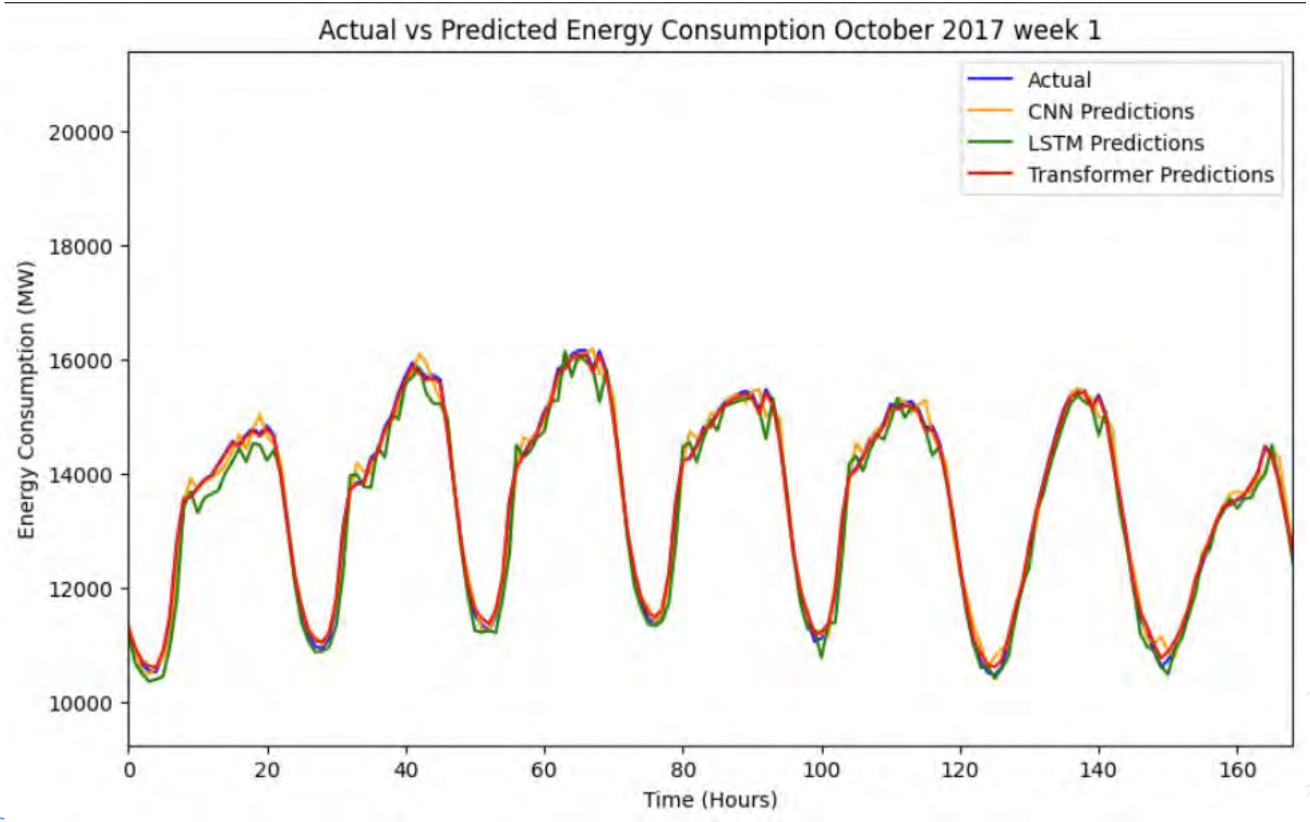
Visualization of all 3 models



Visualization of all 3 models



Visualization of all 3 models



Conclusion

Transformer models ability to use:

- Attention Mechanism
- Parallel processing
- Scalability
- Flexibility
- Long-range Dependencies

Future improvements

Future implementation of new and challenging dataset will make this project more advanced and improve the way we consume our energy.

Reference

Bryant, M. (n.d.). Electric vehicle charging dataset [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/michaelbryantds/electric-vehicle-charging-dataset

Robikscube. (n.d.). Hourly energy consumption [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption

Qingsong. (n.d.). Time-series transformers: A comprehensive review [GitHub repository]. GitHub. Retrieved from https://github.com/qingsongedu/time-series-transformers-review

Intel Tech. (2021, April 30). How to apply transformers to time-series models: Spacetimeformer. Medium. Retrieved from https://medium.com/intel-tech/how-to-apply-transformers-to-time-series-models-spacetimeformer-e452f2825d2e

Hugging Face. (n.d.). Time series transformer. Retrieved from https://huggingface.co/docs/transformers/en/model_doc/time_series_transformer